

Foundation of Cryptography, Lecture 7&8

Interactive Proofs and Zero Knowledge¹

Handout Mode

Iftach Haitner

Tel Aviv University.

December 10&17, 2020

¹ Prepared: 2025/12/18, 14:57:47.

Part I

Interactive Proofs

\mathcal{NP} as a Non-interactive Proofs

Definition 1 (\mathcal{NP})

$\mathcal{L} \in \mathcal{NP}$ iff \exists and poly-time algorithm V such that:

- ▶ $\forall x \in \mathcal{L}$ there exists $w \in \{0, 1\}^*$ s.t. $V(x, w) = 1$
- ▶ $V(x, w) = 0$ for every $x \notin \mathcal{L}$ and $w \in \{0, 1\}^*$

Only $|x|$ counts for the running time of V .

A proof system

- ▶ Efficient verifier, efficient prover (given the witness)
- ▶ Soundness holds unconditionally

Interactive proofs

Protocols between **efficient** verifier and **unbounded** provers.

Definition 2 (Interactive proof)

A protocol (P, V) is an **interactive proof** for \mathcal{L} , if V is PPT and:

Completeness $\forall x \in \mathcal{L}, \Pr[(P, V)(x) = 1] \geq 2/3$.

Soundness $\forall x \notin \mathcal{L}$, and **any** algorithm P^*
 $\Pr[(P^*, V)(x) = 1] \leq 1/3$.

IP is the class of languages that have interactive proofs.

- ▶ $\text{IP} = \text{PSPACE}$!
- ▶ We typically consider (and achieve) perfect completeness.
- ▶ Negligible “soundness error” achieved via repetition.
- ▶ Sometime we have efficient provers via “auxiliary input”.
- ▶ Relaxation: *Computationally sound proofs* [also known as, *interactive arguments*]: soundness only guaranteed against **efficient** (PPT) provers.

Section 1

Interactive Proof for Graph Non-Isomorphism

Graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are **isomorphic**, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.

- ▶ $\mathcal{GI} = \{(G_0, G_1) : G_0 \equiv G_1\} \in \mathcal{NP}$
- ▶ Does $\mathcal{GNI} = \{(G_0, G_1) : G_0 \not\equiv G_1\} \in \mathcal{NP}$?
- ▶ We will show a simple interactive proof for \mathcal{GNI}
Idea: Beer tasting...

Interactive proof for \mathcal{GNI}

Protocol 4 $((P, V))$

Common input: $G_0 = ([m], E_0), G_1 = ([m], E_1)$.

1. V chooses $b \xleftarrow{R} \{0, 1\}$ and $\pi \xleftarrow{R} \Pi_m$, and sends $\pi(E_b)$ to P .^a
2. P send b' to V (tries to set $b' = b$).
3. V accepts iff $b' = b$.

$$^a \pi(E) = \{(\pi(u), \pi(v)) : (u, v) \in E\}.$$

Claim 5

The above protocol is IP for \mathcal{GNI} , with perfect completeness and soundness error $\frac{1}{2}$.

Proving Claim 5

- ▶ Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)
- ▶ $([m], \pi(E_i))$ is a random element in $[G_i]$ — the equivalence class of G_i

Hence,

$$G_0 \equiv G_1: \Pr[b' = b] \leq \frac{1}{2}.$$

$$G_0 \not\equiv G_1: \Pr[b' = b] = 1 \text{ (i.e., } P \text{ can, possibly inefficiently, extract from } \pi(E_i))$$

□

Part II

Zero knowledge Proofs

Where is Waldo?



Question 6

Can you prove you know where Waldo is **without** revealing his location?

The concept of zero knowledge

- ▶ Proving w/o revealing any additional information.
- ▶ What does it mean?

Simulation paradigm.

Protocols, notation

Let $\Pi = (A, B)$ be a two-party protocol in which each party has a private input, and the parties has a common input.

- ▶ $\langle (A(a), B(b))(x) \rangle$: the parties' join view of the in a random execution of Π , in which A has input a , B has input b , and common input x .

The randomness is over the parties coins.

- ▶ For $P \in \{A, B\}$, $\langle (A(a), B(b))(x) \rangle_P$ denote P 's part of the view in $\langle (A(a), B(b))(x) \rangle$.

Distribution ensembles, revisited

We will consider distribution ensembles indexed by arbitrary sets.

Let $\mathcal{L} \subseteq \{0, 1\}^*$, and let $P = \{P_x\}_{x \in \mathcal{L}}$ and $Q = \{Q_x\}_{x \in \mathcal{L}}$ be two distribution ensembles.

P is computationally indistinguishable from Q , denoted $P \approx_c Q$, means that

$$\left| \Pr_{y \leftarrow P_x} [D(x, y) = 1] - \Pr_{y \leftarrow Q_x} [D(x, y) = 1] \right| \leq \text{neg}(|x|)$$

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

An interactive proof (P, V) is **computational zero-knowledge** (\mathcal{CZK}) for \mathcal{L} , if \forall PPT V^* , \exists PPT S (i.e., simulator) such that

$$\{ \langle (P, V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ S(x) \}_{x \in \mathcal{L}} \quad (1)$$

Perfect \mathcal{ZK} (\mathcal{PZK})/**statistical \mathcal{ZK}** (\mathcal{SZK}) — the above distributions are identically/statistically close.

1. \mathcal{ZK} is a property of the **prover**.
2. \mathcal{ZK} only required to hold wrt. **true** statements.
3. If P takes input $w(x) \in \mathcal{R}_{\mathcal{L}}(x)$, we consider $\langle (P(w(x)), V^*)(x) \rangle_{V^*}$
4. Trivial to achieve for $\mathcal{L} \in \mathcal{BPP}$.
5. The \mathcal{NP} proof system is typically **not** zero knowledge.
6. Meaningful also for languages outside \mathcal{NP} .
7. Auxiliary input (will give formal def later)

Zero-knowledge proofs, cont.

1. Security parameter
2. ZK for **honest verifiers**: (1) only holds for $V^* = V$.
3. We sometimes assume for notational convenient, and wlg, that a cheating V^* **outputs** its view.
4. **Statistical** ZK proofs are believed to exist only for a restricted subclass of \mathcal{NP} , so to go beyond that we settle for **computational** ZK (as in this course) or for arguments.
5. Weaker variants: **witness hiding** and **witness indistinguishability**

Section 2

Zero-Knowledge Proof for Graph Isomorphism

Zero-knowledge proof for \mathcal{GI}

Idea: route finding

Protocol 8 ((P, V))

Common input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

P's input: a permutation π over $[m]$ such that $\pi(E_1) = E_0$.

1. P chooses $\pi' \leftarrow \Pi_m$ and sends $E = \pi'(E_0)$ to V.
2. V sends $b \leftarrow \{0, 1\}$ to P.
3. If $b = 0$, P sets $\pi'' = \pi'$, otherwise, it sends $\pi'' = \pi' \circ \pi$ to V.
4. V accepts iff $\pi''(E_b) = E$.

Claim 9

Protocol 8 is a \mathcal{SZK} for \mathcal{GI} , with perfect completeness and soundness $\frac{1}{2}$.

Proving Claim 9

- ▶ Completeness: Clear
- ▶ Soundness: If exist $j \in \{0, 1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.

Assuming V rejects w.p. less than $\frac{1}{2}$ and let π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).

Then $\pi_0^{-1}(\pi_1(E_1)) = \pi_0 \implies (G_0, G_1) \in \mathcal{GI}$.

- ▶ \mathcal{ZK} :
 - ▶ Honest verifier?
 - ▶ Arbitrary verifier? for $(G_0, G_1) \in \mathcal{GI}$, it is easy to generate a random transcript for Steps 1–2, and to be able to open it with prob $\frac{1}{2}$.

The simulator

For a start, consider a deterministic cheating verifier V^* that never aborts.

Algorithm 10 (S)

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

Do $|x|$ times:

1. Choose $b' \leftarrow \{0, 1\}$ and $\pi \leftarrow \Pi_m$, and “send” $\pi(E_{b'})$ to $V^*(x)$.
2. Let b be V^* ’s answer. If $b = b'$, send π to V^* , output V^* ’s view and halt. Otherwise, **rewind** V^* to its initial step, and go to step 1.

Abort.

Claim 11

$$\{ \langle (P, V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx \{ S(x) \}_{x \in \mathcal{L}}$$

Claim 11 implies that **Protocol 8** is zero knowledge.

Proving Claim 11

Consider the following inefficient simulator:

Algorithm 12 (S')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$.

Do $|x|$ times:

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $V^*(x)$.
2. Let b be V^* 's answer.

W.p. $\frac{1}{2}$,

2.1 Find π' such that $E = \pi'(E_b)$, and send it to V^* .

2.2 Output V^* 's view and halt.

Otherwise, rewind V^* to its initial step, and go to step 1.

Abort.

Claim 13

$S(x) \equiv S'(x)$ for any $x \in \mathcal{GI}$.

Proof: ?

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S'')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $V^*(x)$.
2. Find π' such that $E = \pi'(E_b)$ and send it to V^*
3. Output V^* 's view and halt.

Claim 15

$\forall x \in \mathcal{GI}$:

1. $\langle (P, V^*)(x) \rangle_{V^*} \equiv S''(x)$.
2. $SD(S''(x), S'(x)) \leq 2^{-|x|}$.

Proof: ? (1) is clear.

Proving Claim 15(2)

Fix $t \in \{0, 1\}^*$ and let $\alpha = \Pr_{S''(x)}[t]$.

It holds that

$$\begin{aligned}\Pr_{S'(x)}[t] &= \alpha \cdot \sum_{i=1}^{|x|} \left(1 - \frac{1}{2}\right)^{i-1} \cdot \frac{1}{2} \\ &= (1 - 2^{-|x|}) \cdot \alpha\end{aligned}$$

Hence, $SD(S''(x), S'(x)) \leq 2^{-|x|} \square$

Remarks

1. Perfect \mathcal{ZK} for “expected polynomial-time” simulators.
2. Aborting verifiers.
3. Randomized verifiers.
 - 3.1 The simulator first **fixes** the coins of V^* at random.
 - 3.2 Same proof goes through.
4. Negligible soundness error?
 - 4.1 Amplify by repetition
 - 4.2 But what about the \mathcal{ZK} ?

“Transcript simulation” might not suffice!

Let (G, E, D) be a public-key encryption scheme and let $\mathcal{L} \in \mathcal{NP}$.

Protocol 16 $((P, V))$

Common input: $x \in \{0, 1\}^*$

P's input: $w \in \mathcal{R}_{\mathcal{L}}(x)$

1. V samples $(d, e) \xleftarrow{R} G(1^{|x|})$ and sends e to P
2. P sends $c = E_e(w)$ to V
3. V accepts iff $D_d(c) \in \mathcal{R}_{\mathcal{L}}(x)$

- ▶ The above protocol has perfect completeness and soundness.
- ▶ Is it zero-knowledge?
- ▶ It has “transcript simulator” (at least for honest verifiers): \exists PPT S s.t.:
$$\{ \langle (P(w \in \mathcal{R}_{\mathcal{L}}(x)), V)(x) \rangle_{\text{trans}} \}_{x \in \mathcal{L}} \approx_c \{ S(x) \}_{x \in \mathcal{L}},$$

where *trans* stands for the transcript of the protocol (i.e., the *messages* exchange through the execution).

Section 3

Composition of Zero-Knowledge Proofs

Is zero-knowledge maintained under composition?

- ▶ Auxiliary-input zero-knowledge, see next, is maintained under **sequential** repetition.
- ▶ Zero-knowledge might not maintained under **parallel** repetition (and there seems to be no syntactic way to solve it).

Examples:

- ▶ Chess game
- ▶ Signature game

Zero-knowledge proof, auxiliary input variant

Definition 17 (zero-knowledge proofs, auxiliary input)

An interactive proof (P, V) is **auxiliary-input** computational zero-knowledge (\mathcal{CZK}) for \mathcal{L} , if \forall **deterministic** poly-time V^* , \exists PPT S s.t.

$$\{ \langle (P, V^*(z(x)))(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ S(x, z(x)) \}_{x \in \mathcal{L}}.$$

for any poly-output $z: \mathcal{L} \mapsto \{0, 1\}^*$.

Perfect \mathcal{ZK} (\mathcal{PZK})/statistical auxiliary-input \mathcal{ZK} (\mathcal{SZK}) — the above distributions are identically/statistically close.

- ▶ Strengthening of the standard definition.
- ▶ The protocol for \mathcal{GI} we just saw, is also auxiliary-input \mathcal{SZK}
- ▶ What about randomized verifiers?
- ▶ Necessary for proving that zero-knowledge proof compose sequentially.
- ▶ To keep things simple, we will typically prove the non-auxiliary zero-knowledge, but all proofs we present can easily modified to achieve the stronger auxiliary input variant.

Is non-auxiliary-input ZK is auxiliary input ZK?

- ▶ Let $\mathcal{L} = \{1^n : n \in \mathbb{N}\}$ and consider the \mathcal{NP} -relation for \mathcal{L} defined as $\mathcal{R}_{\mathcal{L}} = \{(1^n, 0), (1^n, 1) : n \in \mathbb{N}\}$.
- ▶ Assume exists commitment scheme Com , that is computationally hiding against PPT *uniform* receivers, but **not** hiding against *non-uniform* PPT receivers.
- ▶ The following protocol is \mathcal{CZK} , but **not** auxiliary-input \mathcal{CZK} for \mathcal{L} (it is not even witness hiding).

Protocol 18 ((P, V))

Common input: $x \in \{0, 1\}^*$

P's input: $w \in \mathcal{R}_{\mathcal{L}}(x)$

1. P commits to w using $\text{Com}(1^{|x|})$
2. V accepts if $x \in \mathcal{L}$.

Section 4

Black-box Zero Knowledge

Black-box simulators

Definition 19 (Black-box simulator)

(P, V) is \mathcal{CZK} with black-box simulation for \mathcal{L} , if \exists oracle-aided PPT S :

$$\{\langle (P, V^*(z(x)))(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{S^{V^*(x, z(x))}(x)\}_{x \in \mathcal{L}}$$

for any det. poly-time V^* , and poly-output $z: \mathcal{L} \mapsto \{0, 1\}^*$.

Perfect and statistical variants are defined analogously.

1. "Most simulators" are black box
2. Strictly weaker than general simulation!

Section 5

Zero-knowledge proofs for all NP

\mathcal{CZK} for 3COL

- ▶ Assuming OWFs exists, we give a (black-box) \mathcal{CZK} for 3COL .
- ▶ We show how to transform it for any $\mathcal{L} \in \mathcal{NP}$ (using that $3\text{COL} \in \mathcal{NPC}$).

Definition 20 (3COL)

$G = (M, E) \in 3\text{COL}$, if $\exists \phi: M \mapsto [3]$ s.t. $\phi(u) \neq \phi(v)$ for every $(u, v) \in E$.

We use [commitment schemes](#).

The protocol

Let π_3 be the set of all permutations over $[3]$. We use perfectly binding commitment $\text{Com} = (\text{Snd}, \text{Rcv})$.

Protocol 21 ((P, V))

Common input: graph $G = (M, E)$.

P's input: a (valid) coloring ϕ of G

1. P chooses $\pi \leftarrow \Pi_3$ and sets $\psi = \pi \circ \phi$
2. $\forall v \in M$: parties interact in $(\text{Snd}(\psi(v)), \text{Rcv})(1^{|G|})$.
Let c_v and d_v be the resulting commitment and decommitment.
3. V sends $e = (u, v) \leftarrow E$ to P
4. P sends $(d_u, \psi(u)), (d_v, \psi(v))$ to V
5. V verifies that
 - 5.1 Both decommitments are valid,
 - 5.2 $\psi(u), \psi(v) \in [3]$, and
 - 5.3 $\psi(u) \neq \psi(v)$.

Claim 22

The above protocol is a \mathcal{CZK} for 3COL , with perfect completeness and soundness error $1 - 1/|E|$.

- Completeness: Clear
- Soundness: Let $\{c_v\}_{v \in M}$ be the commitments resulting from an interaction of V with an arbitrary P^* .

Define $\phi: M \mapsto [3]$ as follows:

$\forall v \in M$: let $\phi(v)$ be the (single) value that it is possible to decommit c_v into (if not in $[3]$, set $\phi(v) = 1$).

If $G \notin 3\text{COL}$, then $\exists (u, v) \in E$ s.t. $\phi(u) \neq \phi(v)$.

Hence, V rejects such x w.p. at least $1/|E|$.

Proving \mathcal{ZK}

Fix a deterministic, non-aborting V^* that gets no auxiliary input.

Algorithm 23 (S)

Input: $G = (M, E)$.

Do $|G| \cdot |E|$ times:

1. Choose $e' = (u, v) \leftarrow E$.

1.1 Set $\psi(u) \leftarrow [3]$,

1.2 Set $\psi(v) \leftarrow [3] \setminus \{\psi(u)\}$, and

1.3 Set $\psi(w) = 4$ for $w \in M \setminus \{u, v\}$.

2. $\forall v \in M$: commit to $\psi(v)$ to V^* (resulting in c_v and d_v)

3. Let e be the edge sent by V^* .

If $e = e'$, send $(d_u, \psi(u)), (d_v, \psi(v))$ to V^* , output V^* 's view and halt.

Otherwise, rewind V^* to its initial step, and go to step 1.

Abort.

Proving \mathcal{ZK} cont.

Algorithm 24 (\tilde{S})

Input: $G = (M, E)$, and a (valid) coloring ϕ of G .

Do for $|G| \cdot |E|$ times:

1. Choose $e' \leftarrow E$.
2. Act like the honest prover does given private input ϕ .
3. Let e be the edge sent by V^* . If $e = e'$
 - 3.1 Send $(\psi(u), d_u), (\psi(v), d_v)$ to V^* ,
 - 3.2 Output V^* 's view and halt.

Otherwise, **rewind** V^* to its initial step, and go to step 1.

Abort.

Claim 25

$$\{ \langle (P(w(x)), V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx \{ \tilde{S}^{V^*}(x)(x, w(x)) \}_{x \in \mathcal{L}}$$

Proof: ?

Proving \mathcal{ZK} cont..

Claim 26

$\{S^{V^*}(x)(x)\}_{x \in \mathcal{L}} \approx_c \{\tilde{S}^{V^*}(x)(x, w(x))\}_{x \in \mathcal{L}}$, for any w with $w(x) \in \mathcal{R}_{\mathcal{L}}(x)$.

Proof: Assume $\exists (x, w) \in \mathcal{R}_{\mathcal{L}}$, PPT D , $p \in \text{poly}$ and an infinite set $\mathcal{I} \subseteq \mathcal{L}$ s.t.

$$\Pr \left[D(S^{V^*}(x)(x)) = 1 \right] - \Pr \left[D(\tilde{S}^{V^*}(x)(x, w)) = 1 \right] \geq \frac{1}{p(|x|)}$$

for all $x \in \mathcal{L}$.

Hence, \exists PPT R^* and $b \in [3]$ such that

$$\begin{aligned} & \Pr \left[\left\langle (\text{Snd}(4), R^*(x, w)) (1^{|x|}) \right\rangle_{R^*} = 1 \right] - \Pr \left[\left\langle (\text{Snd}(b), R^*(x, w)) (1^{|x|}) \right\rangle_{R^*} = 1 \right] \\ & \geq \frac{1}{|x| \cdot p(|x|)} \end{aligned}$$

for all $x \in \mathcal{I}$. In contradiction to the (non-uniform) security of Com .

Remarks

- ▶ Aborting verifiers
- ▶ Auxiliary inputs
- ▶ Soundness amplification

Extending to all \mathcal{NP}

For $\mathcal{L} \in \mathcal{NP}$, let Map_X and Map_W be two poly-time computable functions s.t.

- ▶ $x \in \mathcal{L} \iff \text{Map}_X(x) \in 3\text{COL}$
- ▶ $w \in \mathcal{R}_{\mathcal{L}}(x) \iff \text{Map}_W(w) \in \mathcal{R}_{3\text{COL}}(\text{Map}_X(x))$.

Let (P, V) be a \mathcal{CZK} for 3COL with black-box simulation.

Protocol 27 $((P_{\mathcal{L}}, V_{\mathcal{L}}))$

Common input: $x \in \mathcal{L}$.

$P_{\mathcal{L}}$'s input: $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. The two parties interact in $(P(\text{Map}_W(w)), V(\text{Map}_X(x)))$,
where $P_{\mathcal{L}}$ and $V_{\mathcal{L}}$ taking the role of P and V respectively.
2. $V_{\mathcal{L}}$ accepts iff V accepts in the above execution.

Claim 28

$(P_{\mathcal{L}}, V_{\mathcal{L}})$ is a \mathcal{CZK} for \mathcal{L} with the same completeness and soundness as (P, V) as for 3COL .

Completeness and soundness are clear (?)

Proving zero knowledge of $(P_{\mathcal{L}}, V_{\mathcal{L}})$

- ▶ Let S be a black-box simulator of (P, V) .
- ▶ The oracle-aided $S_{\mathcal{L}}$ is defined by $S_{\mathcal{L}}^{(\cdot)}(x) := S^{(\cdot)}(\text{Map}_X(x))$.

Claim 29

\forall poly-time $V_{\mathcal{L}}^*$ and $(x, w) \in \mathcal{R}_{\mathcal{L}}$:

$$\{\langle (P_{\mathcal{L}}(w_{\mathcal{L}}(x)), V_{\mathcal{L}}^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x)\}_{x \in \mathcal{L}}$$

Proof:

- ▶ Assume for simplicity that Map_X is injective.
- ▶ Let $w_{\mathcal{L}}$ be some witness function for \mathcal{L} .
- ▶ Assume $\{\langle (P_{\mathcal{L}}(w_{\mathcal{L}}(x)), V_{\mathcal{L}}^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \not\approx_c \{S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x)\}_{x \in \mathcal{L}}$.

$$\Rightarrow \{\langle (P(\text{Map}_W(w_{\mathcal{L}}(x)), V^*)(\text{Map}_X(x))) \rangle_{V^*} \}_{x \in \mathcal{L}} \not\approx_c \{S^{V^*}(\text{Map}_X(x))\}_{x \in \mathcal{L}}$$

for $V^*(x) := V_{\mathcal{L}}^*(\text{Map}_X^{-1}(x))$.

$$\Rightarrow \{\langle (P(w(x)), V^*)(x) \rangle_{V^*} \}_{x \in 3\text{COL}} \not\approx_c \{S^{V^*(x)}(x)\}_{x \in 3\text{COL}}$$

for the appropriate witness function w of 3COL .

Part III

Proof of Knowledge

Proof of Knowledge

The protocol (P, V) is a **proof of knowledge** for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x , then P^* “knows” $w \in \mathcal{R}_{\mathcal{L}}(x)$.

Definition 30 (Knowledge extractor)

Let (P, V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a **knowledge extractor** for (P, V) and $\mathcal{R}_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t.

$\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in \mathcal{R}_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

(P, V) is a proof of knowledge for \mathcal{L} with error η .

- ▶ A property of V
- ▶ Why do we need it? Authentication schemes
- ▶ Randomized P^* ?

Examples

Claim 31

The \mathcal{ZK} proof we've seen in class for \mathcal{GI} , has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Claim 32

The \mathcal{ZK} proof we've seen in class for 3COL , has a knowledge extractor with error $1 - \frac{1}{|E|}$.

Proof: ?

Part IV

Schnorr Proofs

The settings

- ▶ Fix a multiplicative group ensemble $\mathcal{G} = \{\mathcal{G}_n\}_{n \in \mathbb{N}}$, where each \mathcal{G}_n is a (cyclic) group of prime order p_n .
- ▶ wlg. the \mathcal{G}_n 's do not intersect. (?)
- ▶ \mathcal{G} is **efficient**: computing the order p_n , the group operations (multiplication and inverse) in \mathcal{G}_n , and operation over the field \mathbb{F}_{p_n} are computable in time $\text{poly}(n)$.
- ▶ We will view \mathcal{G} also as a language.
- ▶ Let $\mathcal{R}_{\mathcal{G}} := \{(x, (n, G, X)) : G, X \in \mathcal{G}_n \wedge G^x = X\}$
- ▶ For ease of notation, we will mostly focus on a single n , and omit it from the notation
- ▶ Scalar operations are carried in \mathbb{F}_p .
- ▶ Note that $(G^x)^{-1} = G^{-x}$.

Efficient ZK-POK

- ▶ Given $X = G^x$, we would like to prove in ZK-POK the knowledge of x .
- ▶ Since an \mathcal{NP} statement, we could use generic tools, but would like to have a more efficient proof.

The ZK-POK protocol

Protocol 33 ((P, V))

Common input: $G, X = G^x$.

P's input: x .

1. P: send $A = G^a$ for $a \xleftarrow{R} \mathbb{Z}_p$ to V.
2. V: send $e \xleftarrow{R} \mathbb{F}_p$ to P.
3. P: send $z \leftarrow ex + a$ to V
4. V: accept if $G^z = X^e \cdot A$.

Claim 34

(P, V) is semi-honest ZK for \mathcal{G} , with knowledge extractor for $\mathcal{R}_{\mathcal{G}}$ of error $1/p$.

Correctness:

$$G^z = G^{ex+a} = G^{ex} \cdot G^a = X^e \cdot A.$$

Semi-honest zero knowledge

Algorithm 35 (S)

Input: $G, X = G^x$.

1. $z, e \xleftarrow{R} \mathbb{F}_p$
2. $A \leftarrow G^z / X^e$.
3. Output (A, e, z) .

1. In the real and emulated executions, (e, z) are identically distributed
2. In both executions, A is the same deterministic function of (X, e, z) .
3. Hence, the protocol is perfect semi-honest ZK for \mathcal{G} .

Special soundness

1. We will prove **special soundness**: there exists an efficient algorithm that given two accepting transcripts (A, e_0, z_0) and (A, e_1, z_1) with $e_0 \neq e_1$, outputs x .
2. Implies the claimed POK (hw).
3. Let $e \leftarrow e_1 - e_0$ and $z \leftarrow z_1 - z_0$ and $x \leftarrow z \cdot e^{-1}$.
- 4.

$$\begin{aligned} G^x &= G^{(z_1 - z_0) \cdot e^{-1}} = (X^{e_1} A \cdot (X^{e_0} A)^{-1})^{e^{-1}} \\ &= (X^{e_1 - e_0})^{e^{-1}} = G^{x(e_1 - e_0) \cdot e^{-1}} \\ &= X. \end{aligned}$$

ElGamal commitments

Is $X = G^x$ a good commitment scheme?

Let G_n be a fixed generator for \mathcal{G}_n (assume it can be found efficiently (given 1^n)).

Definition 36 (ElGamal commitments (G, S, R))

- ▶ $G(1^n)$: Output $E \leftarrow \mathcal{G}_n$.
- ▶ $S_E(1^n, m \in \mathbb{F}_{p_n})$: Output $(G_n^r, G_n^m \cdot E^r)$ for $r \xleftarrow{R} \mathbb{F}_{p_n}$.
- ▶ Hiding and binding are defined as usual, but with respect to to an honest key generator algorithm.
- ▶ Similar to ElGamal encryption but the message is in the exponent.
- ▶ What if $m \notin \mathbb{F}_{p_n}$?
- ▶ ElGamal is perfectly binding, and hiding under the right hardness assumption (HW)
- ▶ **Additively homomorphic**: $S_E(m_0; r_0) \cdot S_E(m_1; r_1) = S_E(m_0 + m_1; r_0 + r_1)$
(\cdot stands for point-wise multiplication)

ZK-POK protocol for EG commitments

Protocol 37 ((P, V))

Common input: $G, E, X = S_E(m; r)$.

P's input: m, r .

1. P: Send $A \leftarrow S_E(a; r')$ for $a, r' \xleftarrow{R} \mathbb{Z}_p$ to V.
2. V: Send $e \xleftarrow{R} \mathbb{Z}_p$ to P.
3. P: Send $(z \leftarrow ex + a, r'' \leftarrow er + r') \bmod p$ to V
4. V: Accept iff $S_E(z; r'') = X^e \cdot A$.

Let $\mathcal{R} := \{((m, r), (G, E, S_{G,E}(m; r)))\}$.

Claim 38

(P, V) is semi-honest ZK for $\mathcal{L}(\mathcal{R})$, with knowledge extractor for \mathcal{R} of error $1/p$.

Proof: HW

Part V

Succinct Interactive Arguments

Succinct interactive arguments

Theorem 39

Assume collision-resistant family (CRH) exists, then $\forall \mathcal{L} \in \mathcal{NP}$ exists 4-message, public-coin, interactive argument that on security parameter 1^κ and input $x \in \mathcal{L}$, the parties communicate $O(\kappa \cdot \log^2(|x|))$ bits.

- ▶ Prover is efficient given the witness.
- ▶ Protocol can be made ZK and POK.

PCP theorem

Theorem 40 (PCP theorem, informal)

For every $\mathcal{L} \in \mathcal{NP}$ exists a *one-message* interactive proof (P, V) with perfect completeness s.t. for any $(x, w) \in \mathcal{R}_{\mathcal{L}}$:

1. P is efficient (poly-time) given w .
2. V reads $O(\log^2(|x|))$ *random* locations in the proof.

Commitment with local decommitment

Definition 41 (Commitment with local decommitment)

An efficient two-stage protocol (S, R) :

- ▶ **Commit.** S has private input $\sigma \in \{0, 1\}^*$ and the common input is 1^κ . The commitment stage results in a joint output c , the **commitment**, and a private output d to S .
- ▶ **Local opening.** $S(d, i)$ sends (i, σ_i, ℓ) to R , and R either accepts or rejects.
- ▶ **Completeness.** R always accepts in honest execution, $\forall i \in [|\sigma|]$.
- ▶ **Binding.** With save but $\text{neg}(\kappa)$ probability, a non-uniform PPT sender S^* cannot make the receiver accept two local openings (for same index i).
- ▶ Interesting if $\ell \ll |\sigma|$.
- ▶ No hiding requirement.

Commitment with local decommitment from CRH

Definition 42 (collision resistant hash family (CRH), non-uniform variant)

A function family $\mathcal{H} = \{\mathcal{H}_\kappa: \{0, 1\}^* \mapsto \{0, 1\}^\kappa\}$ is **collision resistant**, if

$$\Pr_{h \leftarrow \mathcal{H}_\kappa} [A(1^\kappa, h) = (x, x') \text{ s.t. } x \neq x' \wedge h(x) = h(x')] = \text{neg}(\kappa)$$

for any non-uniform PPT A .

We assume it takes κ bits to describe $h \in \mathcal{H}_\kappa$.

Theorem 43

Assume CRH exist, then exists two-message, public-coin, commitment with local decommitment, with commitment length κ and local opening length $O(\kappa \cdot \log |\sigma|)$.

Proof: Via **Merkle tree** (board).

Proving Thm 39

Let $\mathcal{L} \in \mathcal{NP}$, $(P_{\text{PCP}}, V_{\text{PCP}})$ be a PCP for \mathcal{L} and (S, R) be a commitment with local decommitment.

Protocol 44 $((P(w), V)(1^\kappa, x))$

1. P compute $\pi = P_{\text{PCP}}(x, w)$.
2. The parties interact in $(S(\pi), R)(1^\kappa)$.
3. V sends the queries of $V_{\text{PCP}}(x)$ to P .
4. P locally decommits the required locations in π .
5. V verifies the openings. If all valid, outputs V_{PCP} 's decision on them.

Soundness proof? Given a cheating prover P^* that makes V accept $x \notin \mathcal{L}$:

1. Extract a proof π^* from P^* .
2. Use π^* to break the soundness of $(P_{\text{PCP}}, V_{\text{PCP}})$.

Making the protocol ZK

Protocol 45 $((P_{ZK}(w), V_{ZK})(1^\kappa, x))$

In each round i , rather than sending the message m_i , the prover

1. Commits to m_i using a **perfectly binding** commitment.
2. Proves, in ZK-POK, it knows the opening.

After the protocol ends, the prover proves in a ZK proof that the values in the commitments make V accept (on the same challenges).

- ▶ Soundness? In each round, extract message from P_{ZK}^* and send it to V .
- ▶ Zero knowledge? Commit to garbage, and simulate the zero knowledge proofs.